METHOD FOR INSERTING AUXILIARY DATA IN AN AUDIO DATA STREAM

The present invention relates to embedding of data or synchronisation signals in another data stream. The invention is particularly concerned with inserting information into a data stream which has been or is intended to be coded, particularly compressed, a particular example being from a linear digital format such as PCM to an MPEG (or similar) audio bitstream. Details of MPEG audio coding are defined in ISO/IEC standards IS 11172-3 and IS 13818-3.

WO-A-98/33284, the disclosure of which is incorporated herein by reference, describes a method of audio signal processing in which auxiliary data is communicated with a decoded audio signal to assist in subsequent re-encoding of the audio signal. Several methods of communicating the data are disclosed; however, the inventor has found that there is room for improvement of the methods disclosed in that application.

The inventor has appreciated that another application in which it would be useful to carry additional data with an audio bitstream is to establish frame boundaries and synchronisation with a previously coded signal. In particular, WO-A-99/04572, incorporated herein by reference, discloses a method of re-encoding a previously coded signal in which the signal is analysed to determine previous coding characteristics. The inventor has appreciated that if some form of synchronisation information were embedded in the signal, the analysis could be simplified.

There has been discussion of carrying additional data in an audio data signal, for example to carry surround sound information, by inserting the data so as to be nearly imperceptible; proposals of this kind however generally involve complex proprietary signal processing and are not designed to accomodate further coding of the signal.

The invention aims to provide a method of communicating data or synchronisation information together with a main data signal without unduly affecting the transmission of the main data signal.

In a first aspect, the invention provides a method of inserting auxiliary digital data in a main digital data stream which is subsequently to be coded to produce a coded data stream (or which has been decoded from a coded data stream), the method comprising identifying at least one component of the main data stream which will make substantially no contribution to the coded data stream (or which was not present in the coded data stream) and inserting data from the auxiliary data stream in the or each component.

In this way, the eventual coded data stream will be substantially unaffected by the insertion of the auxiliary data, so there will be no overall degradatic or distortion introduced by the extra data. However, the auxiliary data w have been carried "for free" with the main data signal until it reaches th coder. Although the invention will normally be employed in conjunctic data which is to be coded subsequently (in which case the auxillary r may be removed at or around the time of coding), the invention ma employed with data which has previously been coded but is not ne to be coded further; this still provides the advantage that the car additional information cannot degrade the data further as no "re information is overwritten by the auxillary data.

A further potential advantage is that, because the insertion on the principles used in coding, components can be share data insertion apparatus and a coder or decoder, particul complex proprietary signal processing and are not designed to accomodate further coding of the signal.

The invention aims to provide a method of communicating data or synchronisation information together with a main data signal without unduly affecting the transmission of the main data signal.

In a first aspect, the invention provides a method of inserting auxiliary digital data in a main digital data stream which is subsequently to be coded to produce a coded data stream (or which has been decoded from a coded data stream), the method comprising identifying at least one component of the main data stream which will make substantially no contribution to the coded data stream (or which was not present in the coded data stream) and inserting data from the auxiliary data stream in the or each component.

In this way, the eventual coded data stream will be substantially unaffected by the insertion of the auxiliary data, so there will be no overall degradation or distortion introduced by the extra data. However, the auxiliary data will have been carried "for free" with the main data signal until it reaches the coder. Although the invention will normally be employed in conjunction with data which is to be coded subsequently (in which case the auxillary data may be removed at or around the time of coding), the invention may be employed with data which has previously been coded but is not necessarily to be coded further; this still provides the advantage that the carrying of additional information cannot degrade the data further as no “real” information is overwritten by the auxillary data.

A further potential advantage is that, because the insertion of data is based on the principles used in coding, components can be shared between the data insertion apparatus and a coder or decoder, particularly when integrated as a unit including a data insertion function and a coding or decoding function, rather than requiring bespoke design. The auxillary data may be carried further with the coded data stream, but no longer embedded in the main data stream. For example, in the case of coded audio, the coded data format may allow the auxillary data to be carried directly as data in addition to the coded audio. The auxiliary data is preferably used to assist in choosing coding decisions or in synchronising the coder with a previous coder. The main data signal is preferably an audio signal, but may be a video or other signal.

Whilst the invention is primarily concerned with adding information to a digital main data signal, it is to be appreciated that this digital signal can be converted into other forms; for example a linear PCM digital signal carrying embedded digital data or a synchronisation signal may be converted to analogue form and back again and provided the conversion is faithful, the data may be recovered, or at least the synchronisation signal may be identified.

The method may further include extracting the auxillary data and coding the main data. At least one coding parameter or decision is preferably based on the auxillary data.

Preferably coding includes quantising data words corresponding to said main digital data stream or, more preferably, a transformed data stream to a plurality of levels less than the number of levels codable by said data words. The component of the main data stream may corresponds to less significant bits of coded data words which are to be quantised by said coding to one of a predetermined number of levels, the number of levels being less than the number of levels encodable by the data words. For example, if an n-bit word is to be quantised by coding to 2^m levels, where m<n, n-m bits may be

available to carry additional data.

Preferably, the change in the data stream effected by insertion of the auxillary data is substantially imperceptible, for example below (or at) the audible noise floor in the case of audio data or having substantially no perceptible effect on picture quality in the case of a video signal.

Preferably inserting the auxillary data comprises inserting the data into unused sub-band samples of a transformed set of data.

In a preferred application, the main data comprises audio data to be coded according to an MPEG-type audio coding scheme (by which is meant any similar coding scheme based on the principle of quantising a plurality of sub bands or other components into which the signal is analysed) and identifying at least one component comprises identifying sub-bands which are unoccupied or identifying quantisation levels, the auxillary data being inserted in unoccupied bands or at a level below the quantisation noise floor.

This may be provided independently in a related but independent aspect, in which the invention provides a method of inserting auxillary data into an audio data stream to be coded by analysing the audio data into a plurality of sub-bands and quantising the sub-bands, the method comprising estimating sub-bands and quantisation levels for a subsequent or previous coding and inserting the auxillary data at a level substantially below the level of estimated quantisation noise.

Estimating sub-bands and quantisation levels may include transforming the (audio) data from the time domain (or an uncoded domain) to the frequency domain (or a coded domain) or otherwise analysing the data into a plurality of subbands, for example using a Fourier or the like transform. Data may be inserted in the frequency domain, and the modified frequency domain data may be transformed back to the time domain.

A particular advantage arises when the estimated sub bands or quantisation levels correspond directly to sub bands or quantisation parameters which have been or will be used in coding of the data; there is substantially no effect on the coded signal, as the component(s) of the main data signal which are used to carry the auxillary data would otherwise be lost by the coding process.

The data to be carried may comprise a defined synchronisation sequence; this may facilitate detection of frame boundaries and the like and may be employed to facilitate extraction of other data or to minimise degradation between cascaded coding and decoding operations.

The auxillary data or synchronisation signal may be inserted into an upper subband of the main data.

In a further aspect, the invention provides a method of carrying a synchronisation sequence with a main digital data signal, preferably an audio signal, for example a linear PCM audio signal, comprising inserting a defined sequence of synchronisation words into a component of the main data signal, preferably an unused subband, to facilitate identification of or synchronisation with previous coding of the signal.

The invention also provide a method of detecting a frame boundary or establishing synchronisation with a data signal produced by the above method comprising searching for a sequence of synchronisation words in said component of the data signal and comparing at least one value found, or a derived value to a stored sequence of values.

The invention further provides a digital data signal, preferably a linear PCM audio bitstream, comprising an audio signal and at least one of a synchronisation sequence or an auxillary data signal embedded in an otherwise unused subband or in subbands below an MPEG quantisation noise floor.

The invention extends to apparatus for inserting auxillary data into a data stream and to data streams coded by the above method.

Embodiments of the invention will now be described by way of example, with reference to the accompanying drawings in which:

Fig. 1 shows schematically cascaded MPEG-type coding and decoding transformations;

Fig. 2 shows bit allocation for a typical signal;

Fig. 3 shows scalefactors and the lowest level that can be coded for the signal of Fig. 2;

Fig. 4 shows space determined to be available for data transmission in accordance with the invention;

Fig. 5 is an illustration of the effect of 32-sample alignment on an ID sequence

Fig. 6 shows an example synchronisation signal;

Fig. 7 shows insertion and extraction of the synchronisation signal.

A preferred application of the invention involves carrying additional data with an audio signal which is to be coded according to MPEG audio coding. The basic principles will be described, to assist in understanding of the invention.

Carrying data with MPEG audio signals -basic principles

MPEG audio uses the idea of psychoacoustic masking to reduce the amount of information to be transmitted to represent an audio signal. The reduced information is represented as a bitstream. Psychoacoustic masking is usually calculated on a frequency representation of an audio signal. In MPEG audio a filterbank is used to split the audio into 32 subbands, each representing part of the spectrum of the signal.

The encoder uses a psychoacoustical model to calculate the number of bits needed to code each of these subbands such that the quantisation noise inserted is not audible. So, in each subband, only the most significant bits are transmitted.

In this embodiment, the aim is to carry data along with audio in a linear digital PCM form (although other digital formats may be employed). The data should be carried inaudibly and be capable of being fully recovered without data loss. We have found that, depending on the bit-rate used for the MPEG encoding and the nature of the signal, it is possible to transmit between 50 and 400 kbits/sec of data under a stereo audio signal.

General applications of data-carrying possible with the embodiment include carrying associated data with the audio, such as text (e.g. lyrics). In addition, a specific use of the invention, to be described in more detail below, arises if a signal is already in MPEG coded form or has been previously coded but needs to be conveyed in linear form; here the extra data can contain details of the coding process or synchronisation information to assist in subsequent re-coding, or pictures associated with the audio.

The filterbanks in MPEG audio have the property of (nearly) perfect reconstruction. A diagram of a decoder to an encoder is shown in Fig. 1. If the filterbanks are aligned correctly then the subband samples in the encoder will be practically identical to those that originated in the decoder.

When an encoder encodes the signal it attempts to allocate enough bits for each subband such that the resulting signal is not audibly different from the original.

<u>Selection of components for carrying data</u>

Given these two properties, we have appreciated that data can be inserted into the subbands below the level of the significant audio signal such that the inserted data is inaudible (or at least not introducing any impairments beyond those of the MPEG encoding).

Fig. 2 shows the measured level of the audio in each subband, coded as "scalefactors" in the MPEG audio bitstream. It also shows the bit allocation chosen by an encoder. This is specified as the number of quantisation levels for a particular subband. In the diagram, the bit allocation is represented as a signal-to-noise ratio, in dB terms, to permit representation on the same axis. For this purpose, each bit that is needed to represent the number of quatisation levels is approximately equivalent to 6dB of "level".

If instead we show the scalefactors and the lowest level that can be encoded with the bit allocation from Fig. 2 we get the graph in Fig. 3.

One can see that the levels below the lowest level are unused. As the MPEG model has determined that there is no audible information below these lowest levels we are free to use them for data.

Given the constraint that we should not interefere with the audio, levels near that of the lowest level will not be used. This should also mean that no clipping problems are introduced. Given also that the signal is probably to be transmitted or stored over a linear medium with limited resolution (e.g. 16 bits), this imposes a constraint on the lowest level we can send. Due to inaccuracies in reconstruction because of truncation to PCM and limits on accuracy in the filterbank calculation, it is unwise to use the levels closest to the PCM quantisation limit (e.g the 16th bit). In the case of subbands where no information is to be sent two strategies are available.

If we are decoding an MPEG bitstream to insert data, we would not know the level of that subband so, to be safe, we should probably not send any data in that subband. If, on the other hand, we are using an encoder purely for generating data we could use the levels just below the full level in this subband. A diagram showing the area where the data could be inserted, for the latter case, is shown in Fig. 4.

In the case of subbands containing an audio signal, the level of the data will be below the most significant levels. Data could also be inserted into other subbands, below the level of audibility or above the range of normal hearing (e.g. in the subbands not used in MPEG encoding).

Practical Implementation Details

For a practical implementation several issues need to be addressed, in particular how the data is inserted and how the data is recovered. Data could be inserted when decoding an MPEG audio bitstream or the functions of an encoder and decoder could be combined to filter the signal, analyse it, quantise the audio appropriately, insert the data, then convert the signal back to the PCM domain.

Data insertion

A proposed method of data insertion is first to calculate the number of bits available and then mask subband values with the data before they are fed to the synthesis filterbank. A 16-bit system is assumed, but the calculations are similar for a larger number of bits. The scheme described below is simple and safe.

Calculation of the bits available

Take the maximum scalefactor for a subband as representing a maximum value signal that can be conveyed in a 16-bit PCM system. Then consider that approximately 96dB below this is the quantisation floor of the 16-bit PCM system. Scalefactors are defined in 2dB steps. Once the scalefactor for a given subband is calculated determine the difference between this and the noise floor in dB (the range, R). The MPEG psychoacoustic model will give the bit allocation. Translate the bit allocation for the subband to a signal-to-noise figure in dB (Q). Thus calculate the range in dB available for data (D) from the quantisation floor to the lowest level represented.

$$D = R - Q$$

Then subtract the safety margins of 1-bit near the signal and another bit near the noise floor, remembering 1-bit is approximately equivalent to 6dB signal-to-noise.

$$D = D - 12$$

Next allocate a number of data bits (N) per subband by finding the integer number of bits that can be represented within D by doing an integer division on D.

$$N = \text{int}(D / 6)$$

This value is valid for a particular subband and scalefactor. In MPEG Layer 2 there are up to 3 different scalefactors per frame so each could have its own number of bits or the minimum could be taken of all 3 scalefactors.

Masking the data onto the subband value

From the procedure described above the number of bits available (N) is used to create a mask (M).

M = 0xffff << (N+1) for a 16-bit system

The subband value is then converted to a 16-bit integer, masked with this value and the data inserted onto the N Least Significant Bits (excluding the last bit of course) to give a sample S. To ensure the most accurate representation of the signal a rounding value is added to S, +0.5 if the signal is positive and -0.5 if it is negative. This gives almost perfect reconstruction in the analysis filter and the data is recovered perfectly.

An easy method of inserting the data is to treat the data as a bitstream and insert as many bits into each subband as possible. However, to indicate synchronisation it would be useful to put a sequence into consecutive (in time) values of subband values so that a whole frame can be identified.

Data Extraction

To extract the data from the signal, alignment of the filterbanks and a method of describing where the data is (the bit allocation) and how it is organised are needed. These points are addressed below.

Synchronisation

To extract the data, synchronisation with the 32-sample and frame structure of the audio signal are needed. A separate synchronisation signal could be sent or this signal could be included in the data sent. Another possibility is to deduce the 32-sample boundary and then use a synchronisation word within the data to identify the frame boundary. This aspect is discussed further below.

Bit allocation

To extract the data, the position of the data within the subbands must be known. There are several options for how this information is conveyed:

> The bit allocation could be implicit by having the same psychoacoustic model in the receiver of the data as in the transmitter.
>
> The bit allocation could be signalled separately, e.g. in an upper unused subband, in the user bits of an AES/EBU bitstream or by another technique that does not interfere with the system described above.
>
> The bit allocation can be contained within the space for data, with mechanisms provided to signal the location of the bit allocation.

This last option is discussed below.

Data organisation

If the bit allocation is known then the data can be carried in whatever form is suitable for that particular data. A checksum is advisable as well as a synchronisation word to define the start of the frame and/or data. If the bit allocation is to be carried within the data then the dynamic nature of the bit allocation must be taken into account.

An example layout for MPEG Layer 2 audio, using only 1 bit allocation per frame (i.e. not taking into account the 3 possibly different scalefactors) will be discussed.

A synchronisation word is needed to show where the frame starts. This needs to be followed by the bit allocations for each subband, preferably with a checksum and then followed by the data itself, again preferably with a checksum. The synchronisation word should be followed by a pointer to the space where the bit allocation is contained. Due to the dynamic nature of the bit allocation, the following manner of organisation would be appropriate, with the information preferably appearing in the order listed (details may change):

Synchronisation word

This should ideally be placed in the lowest subband with data space available, usually the first subband. The sequence may be placed 1 bit at a time into consecutive (in time) subband values, in the lowest bit available for data transmission. The data receiver may have to search for this word if the sync word is not placed in the first subband. There are a minimum of 36 bits available in a subband per frame and, for example, 18 bits can be used for the sync word.

Pointer to bit allocation

This should point to subbands that have data space available to store the bit allocation. Assuming we use 4 bits per subband to describe the bit-allocation for that subband, with 32 subbands we need 128 bits in total. So, given that we have multiples of 36 bits available per subband per frame, we need to be able to point to areas containing 4 times 36 bits. Given that there are 18 bits available in the synchronisation subband, one possibility is to use a 4-bit pointer to a subband and a 2-bit count of the number of bits available. The 4-bit pointer can indicate an offset upwards to the next subband (with the range 1 to 16). The 2-bit count can be from 1 to 4 bits, as 4 is the maximum number we need. We could then have three of these pointers in the first subband. An exception case could be defined if we only have subbands with 1 bit available.

Bit allocation

This should contain 32 times 4-bits to indicate the number of bits available per subband. It should ideally be followed by a 16-bit checksum to ensure the data is correct, making a total of 144 bits.

The data can then follow the above header information.

The above scheme has an overhead of 180 bits per frame, which is approximately 6900 bits per second per audio channel at 44.1 kHz.

The implementation described above is suitable for carrying whatever data is desired, for example lyrics, graphics or other additional information. Another possibility is, particularly where the data has been previously coded, to carry information on previous coding decisions, for example to reduce impairment in signal quality caused by cascaded decoding and recoding, or to simplify subsequent coding.

A further possibility is to carry a synchronisation signal or data word (in addition to further data or alone) either to assist in establishing synchronisation (as mentioned above) or to facilitate recoding of a previously coded signal by deducing previous coding decisions. An arrangement for carrying a synchronisation signal will now be described.

Carrying a synchronisation signal

The technique to be described below enables deduction of synchronisation from the characteristics of the signal itself, rather than added data. It is also capable of surviving a level change. To assist in understanding, the basic principles of MPEG audio, discussed above, will be summarised again, with reference to this specific implementation.

Synchronisation with MPEG-type audio - Basic Principles

MPEG audio uses a filter to split the audio into different subbands. The PCM input samples are transformed into corresponding subband samples by an analysis filter. These samples are then transformed back into PCM samples by a synthesis filter. There is an inherent delay in this process, dependent on the design of the filterbanks.

For each 32 input PCM samples the analysis filter produces 32 values, one for each subband. This group of subband values is known as a "subband sample". In MPEG audio a fixed number of PCM samples, a frame, are grouped together to make the coding more efficient. MPEG Layer 2, for example, uses a frame length of 1152 PCM samples, which is equivalent to 36 subband samples. Information is then carried in the MPEG bitstream about this whole frame, e.g. the number of bits per subband and the level of each subband as well as the quantised subband values.

The nature of the filterbank is such that when re-encoding a previously encoded signal, the original subband samples will only be recovered if the PCM samples going into the analysis filterbank line up to the same 32-sample boundary as used in the original encoding. If the filterbank 32-sample boundaries are not aligned extra noise will appear in the subbands.

In order to code the audio again optimally it would be useful to know where the 32-sample boundary is, to avoid inserting extra noise. It would also be useful to know where the frame boundary is, so that calculations of the appropriate bit-allocation are based on exactly the same signal. In theory this could lead to transparent re-encoding.

In this application of the invention, the aim is to insert a specific identification sequence into a subband in a decoder, which will then be embedded in the linear PCM output. A subsequent encoder can use this information to deduce the 32-sample boundaries in the original encoding and/or to deduce the frame boundary upon which the original encoding was based.

An advantage of the technique now being described is that deduction is direct from performing a filterbank on the audio. By inserting this identification sequence into an upper subband, the signal will be inaudible and continually present. It could alternatively be inserted into a lower subband, on its own as an identification signal or carried underneath the audio signal. A suitable identification signal could still be decoded after a level change.

Inserting identification sequence

By inserting a suitable identification sequence into a subband, the original values of this sequence will only be recovered exactly when the original 32-sample boundary of the inital analysis filter is matched in the current analysis filterbank. Thus if the PCM audio is offset by something other than 32 samples another unique sequence will be produced. From this the original 32-sample boundaries can be determined. If the sequence is unique across the length of a frame (e.g. 1152 PCM samples for Layer 2, equivalent to 36 consecutive values in 1 particular subband), the frame position can also be easily deduced. An illustrative sequence is shown in Fig. 5.

If a gain change is applied to the PCM audio signal, only the relative levels of the identification sequence will be changed. Thus the same information could still be deduced, dependent on the inserted level of the identification sequence. By careful choice of a suitable identification sequence the frame position can be calculated with only a subset of its 36 samples. The sequence preferably comprises at least 4 words.

Example Identification Sequence

An example synchronisation sequence, shown in Fig. 6, consists of a sine wave with certain points set to zero. This can be inserted into an upper subband, e.g. subband 30. For 48kHz sampling this is above the maximum subband (27) defined by the MPEG standard. Thus this extra synchronisation-signal would not be coded by a "dumb" encoder.

This sequence should be inserted into an appropriate subband before the synthesis filter (see Fig. 7). The analysis filter would then produce subband samples from which the frame and 32-sample boundary can be deduced.

To analyse the offset the modified encoder can use the following simple procedure (assuming it has no synchronisation information at the moment):

Move in the next 32 PCM samples and run the filterbank to obtain a subband sample.

Extract the value from the appropriate subband (e.g. 30).

Check this value against a table of all known possible values for all offsets. (A table of 32 by 36 values.)

If a match has been found, run the filterbank again a couple of times and check the consecutive values in the table.

Derive the exact sample offset required from the position in the table.

When the filterbank is run again with the correct offset, the alignment can be double-checked very easily.

If the synchronisation signal is defined carefully to give unique values for all the offsets and positions the number of comparisons can be kept to a minimum. The synchronisation signal defined above would give a definite answer after running the filterbank 4 times, i.e. with just 4 subband samples. It is possible to define other synchronisation signals which would indicate the delay directly, but there is a trade-off in how much processing power is required to perform the filterbank against the time required for searching tables and deriving values.

A procedure for determining synchronisation when gain has been applied to the signal is similar in principle to the above, but the relative levels of consecutive samples should be used. E.g. if the subband values are A,B,C,... then a table of A/B,B/C,... would be used. This may impose further requirements on the synchronisation signal. The above signal could also indicate if there had been a phase inversion of the audio.

To recap, techniques have been described for carrying data "transparently" in a data stream in a manner which is compatible with subsequent or previous coding, particularly MPEG-type audio coding. Techniques for establishing synchronisation with a previously coded signal have also been described. The invention may be extended to other applications and the preferred features mentioned above may be provided independently unless otherwise stated.